

UNIT23: SKINNED MESH - BONE MATRIX

【学習要項】

□Bone matrices

【演習手順】

1. 前回使用した3つのボーンを持つFBXファイル(cube.004.fbx)をロードしダミー行列をセットしテストする
2. skinned_mesh クラスに定義した定数バッファー(constants)にボーン行列を追加する

```
*1: static const int MAX_BONES{ 256 };
2: struct constants
3: {
4:     DirectX::XMFLOAT4X4 world;
5:     DirectX::XMFLOAT4 material_color;
*6:     DirectX::XMFLOAT4X4 bone_transforms[MAX_BONES]{ { 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1 } };
7: };
```

3. シェーダ(skinned_mesh.hlsl)に定義した定数バッファー(OBJECT_CONSTANT_BUFFER)にボーン行列を追加する

```
*1: static const int MAX_BONES = 256;
2: cbuffer OBJECT_CONSTANT_BUFFER : register(b0)
3: {
4:     row_major float4x4 world;
5:     float4 material_color;
*6:     row_major float4x4 bone_transforms[MAX_BONES];
7: };
```

4. 頂点シェーダで頂点の位置と法線にボーン行列の影響を与える

※頂点シェーダの先頭に下記コードを挿入する

```
1: vin.normal.w = 0;
2: float4 blended_position = { 0, 0, 0, 1 };
3: float4 blended_normal = { 0, 0, 0, 0 };
4: for (int bone_index = 0; bone_index < 4; ++bone_index)
5: {
6:     blended_position += vin.bone_weights[bone_index]
7:         * mul(vin.position, bone_transforms[vin.bone_indices[bone_index]]);
8:     blended_normal += vin.bone_weights[bone_index]
9:         * mul(vin.normal, bone_transforms[vin.bone_indices[bone_index]]);
10: }
11: vin.position = float4(blended_position.xyz, 1.0f);
12: vin.normal = float4(blended_normal.xyz, 0.0f);
```

5. skinned_mesh クラスの render メンバ関数でダミー行列を定数バッファ(data.bone_transforms)にセットする

※動作確認後#if-#endif ディレクティブのコードは無効にすること

```
1:     constants data;
2:     XMStoreFloat4x4(&data.world, XMLoadFloat4x4(&mesh.default_global_transform) * XMLoadFloat4x4(&world));
*3: #if 1
*4:     XMStoreFloat4x4(&data.bone_transforms[0], XMMatrixIdentity());
*5:     XMStoreFloat4x4(&data.bone_transforms[1], XMMatrixRotationRollPitchYaw(0, 0, XMConvertToRadians(+45)));
*6:     XMStoreFloat4x4(&data.bone_transforms[2], XMMatrixRotationRollPitchYaw(0, 0, XMConvertToRadians(-45)));
*7: #endif
```

6. 実行し、メッシュの変形を確認する (スクリーンショット画像: cube.004.1.png)

【評価項目】

□ボーン行列